METHOD AND SYSTEM FOR SYNCHRONIZING COMMUNICATIONS LINKS IN A HUB-BASED MEMORY SYSTEM

TECHNICAL FIELD

This invention relates to computer systems, and, more particularly, to a computer system including a system memory having a memory hub architecture.

BACKGROUND OF THE INVENTION

Computer systems use memory devices, such as dynamic random access memory ("DRAM") devices, to store data that are accessed by a processor. These memory devices are normally used as system memory in a computer system. In a typical computer system, the processor communicates with the system memory through a processor bus and a memory controller. The processor issues a memory request, which includes a memory command, such as a read command, and an address designating the location from which data or instructions are to be read. The memory controller uses the command and address to generate appropriate command signals as well as row and column addresses, which are applied to the system memory. In response to the commands and addresses, data are transferred between the system memory and the processor. The memory controller is often part of a system controller, which also includes bus bridge circuitry for coupling the processor bus to an expansion bus, such as a PCI bus.

Although the operating speed of memory devices has continuously increased, this increase in operating speed has not kept pace with increases in the operating speed of processors. Even slower has been the increase in operating speed of memory controllers coupling processors to memory devices. The relatively slow speed of memory controllers and memory devices limits the data bandwidth between the processor and the memory devices.

In addition to the limited bandwidth between processors and memory devices, the performance of computer systems is also limited by latency problems that increase the time required to read data from system memory devices. More specifically,

when a memory device read command is coupled to a system memory device, such as a synchronous DRAM ("SDRAM") device, the read data are output from the SDRAM device only after a delay of several clock periods. Therefore, although SDRAM devices can synchronously output burst data at a high data rate, the delay in initially providing the data can significantly slow the operating speed of a computer system using such SDRAM devices.

One approach to alleviating the memory latency problem is to use multiple memory devices coupled to the processor through a memory hub. In a memory hub architecture, a system controller or memory controller is coupled over a high speed link to several memory modules. Typically, the memory modules are coupled in a point-to-point or daisy chain architecture such that the memory modules are connected one to another in series. Thus, the memory controller is coupled to a first memory module over a first high speed link, with the first memory module connected to a second memory module through a second high speed link, and the second memory module coupled to a third memory module through a third high speed link, and so on in a daisy chain fashion.

Each memory module includes a memory hub that is coupled to the corresponding high speed links and a number of memory devices on the module, with the memory hubs efficiently routing memory requests and memory responses between the controller and the memory devices over the high speed links. Computer systems employing this architecture can have a higher bandwidth because a processor can access one memory device while another memory device is responding to a prior memory access. For example, the processor can output write data to one of the memory devices in the system while another memory device in the system is preparing to provide read data to the processor. Moreover, this architecture also provides for easy expansion of the system memory without concern for degradation in signal quality as more memory modules are added, such as occurs in conventional multi drop bus architectures.

Signals are transferred over the high speed links at very high rates, with the links being optical, radio frequency, or other suitable high speed communications media. As the data transfer rate increases, the duration for which each signal being

transferred over the high speed link is valid decreases by a corresponding amount, as will be understood by one skilled in the art. More specifically, the data window or "data eye" for each of the signals decreases as the data transfer rate increases. As understood by those skilled in the art, the data eye for each of the signals defines the actual duration for which each signal is valid after timing skew, jitter, duty cycle variation, and other types of unwanted signal distortion are considered. Signal distortion can arise from a variety of sources, such as different loading on the lines of the link and the physical lengths of such lines.

In a conventional system memory, to synchronize memory devices coupled to a memory controller the controller enters an initialization or synchronization mode of operation and applies a test data pattern to the memory devices. Typically, the controller thereafter adjusts the phase of the data strobe signal relative to the signals forming the test data pattern and determines limits for phase shifts of the data strobe signal that allow the memory device to successfully capture the data signals. A phase shift within the determined limits is then selected for use during normal operation of the controller and memory device. In the conventional system memory, each memory device is coupled to the controller over a common memory bus. Conversely, in a memory hub system having a daisy-chain configuration not every memory hub is coupled directly to the controller. The controller does not directly communicate with each memory hub in a memory hub system having more than one memory hub, and therefore the controller cannot synchronize the memory hubs in the same way as in a conventional system memory.

There is a need for a system and method of synchronizing memory hubs in a system memory having a memory hub architecture.

SUMMARY OF THE INVENTION

According to one aspect of the present invention, a method is disclosed for synchronizing communications links in a memory hub system. The memory hub system includes a system controller and a plurality of memory hubs coupled in series, with pairs of downstream and upstream links being coupled between adjacent modules

and the controller. The method includes synchronizing each upstream and downstream link. In a clockwise order starting with the downstream link coupled between the controller and the first memory module, the next adjacent clockwise link is signaled that the prior clockwise link has been synchronized. The method then detects through the upstream link coupled between the controller and the first memory module when all links have been synchronized. In a clockwise order starting with the downstream link coupled between the controller and the first memory module, each link is enabled. The method then detects through the upstream link coupled between the controller and the first memory module when all links have been enabled.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a block diagram of a computer system including a system memory having a high-bandwidth memory hub architecture according to one example of the present invention.

Figure 2 is a block diagram illustrating in more detail the memory hubs contained in the memory modules in the system memory of Figure 1 according to one example of the present invention.

Figure 3 is a functional diagram illustrating the operation of the system controller and the memory modules of Figure 2 during an initialization stage of a synchronization process according to one embodiment of the present invention.

Figure 4 is a functional diagram illustrating the operation of the system controller and memory modules of Figure 2 during an enablement stage of a synchronization process according to one embodiment of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

A computer system 100 according to one example of the present invention is shown in Figure 1. The computer system 100 includes a system memory 102 having a memory hub architecture that executes an initialization or synchronization process to synchronize a plurality of memory hubs 140 contained in a plurality of memory modules 130, as will be explained in more detail below. In the following

description, certain details are set forth to provide a sufficient understanding of the present invention. One skilled in the art will understand, however, that the invention may be practiced without these particular details. In other instances, well-known circuits, control signals, timing protocols, and/or software operations have not been shown in detail or omitted entirely in order to avoid unnecessarily obscuring the present invention.

The computer system 100 includes a processor 104 for performing various computing functions, such as executing specific software to perform specific calculations or tasks. The processor 104 is typically a central processing unit ("CPU") having a processor bus 106 that normally includes an address bus, a control bus, and a data bus. The processor bus 106 is typically coupled to a cache memory 108, which is usually static random access memory ("SRAM"). Finally, the processor bus 106 is coupled to a system controller 110, which is also sometimes referred to as a "North Bridge" or "memory controller."

The system controller 110 serves as a communications path to the processor 104 for the memory modules 130 and for a variety of other components. More specifically, the system controller 110 includes a graphics port that is typically coupled to a graphics controller 112, which is, in turn, coupled to a video terminal 114. The system controller 110 is also coupled to one or more input devices 118, such as a keyboard or a mouse, to allow an operator to interface with the computer system 100. Typically, the computer system 100 also includes one or more output devices 120, such as a printer, coupled to the processor 104 through the system controller 110. One or more data storage devices 124 are also typically coupled to the processor 104 through the system controller 110 to allow the processor 104 to store data or retrieve data from internal or external storage media (not shown). Examples of typical storage devices 124 include hard and floppy disks, tape cassettes, and compact disk read-only memories (CD-ROMs).

The system controller 110 is coupled to the system memory 102 including the memory modules 130a,b…n, and operates to apply commands to control and access data in the memory modules. The system controller 110 also initiates a

synchronization mode of operation of the controller and memory modules 130, as will be explained in more detail below. The memory modules 130 are coupled in a point-to-point or daisy chain architecture through respective high speed links 134 coupled between the modules and the system controller 110. The high-speed links 134 may be optical, RF, or electrical communications paths, or may be some other suitable type of communications paths, as will be appreciated by those skilled in the art. In the event the high-speed links 134 are implemented as optical communications paths, each optical communication path may be in the form of one or more optical fibers, for example. In such a system, the system controller 110 and the memory modules 130 will each include an optical input/output port or separate input and output ports coupled to the corresponding optical communications paths. Although the memory modules 130 are shown coupled to the system controller 110 in a daisy architecture, other topologies that may be used, such as a ring topology, will be apparent to those skilled in the art.

Each of the memory modules 130 includes the memory hub 140 for communicating over the corresponding high-speed links 134 and for controlling access to six memory devices 148, which are synchronous dynamic random access memory ("SDRAM") devices in the example of Figure 1. The memory hubs 140 each include input and output interfaces or ports that are coupled to the corresponding high-speed links 134, with the nature and number of ports depending on the characteristics of the high-speed links. A fewer or greater number of memory devices 148 may be used, and memory devices other than SDRAM devices may also be used. The memory hub 140 is coupled to each of the system memory devices 148 through a bus system 150, which normally includes a control bus, an address bus, and a data bus.

Figure 2 is a block diagram illustrating in more detail the memory hubs in the memory modules 130a and 130b and link interface components in the system controller 110. In the memory module 130a, the memory hub 140 includes a link interface 200 that is connected to the high-speed link 134 coupled to the system controller 110. The link interface 200 includes a downstream physical reception port 202 that receives downstream memory requests from the system controller 110 over a downstream high-speed link 204, and includes an upstream physical transmission port

206 that provides upstream memory responses to the system controller over an upstream high-speed link 208. The downstream and upstream high-speed links 204, 208 collectively form the corresponding high-speed link 134.

The system controller 110 includes a downstream physical transmission port 210 coupled to the downstream high-speed link 204 to provide memory requests to the memory module 130a, and also includes an upstream physical reception port 212 coupled to the upstream high-speed link 208 to receive memory responses from the memory module 130a. The ports 202, 206, 210, 212 and other ports to be discussed below are designated "physical" interfaces or ports since these ports are in what is commonly termed the "physical layer" of a communications system. In this case, the physical layer corresponds to components providing the actual physical connection and communications between the system controller 110 and system memory 102 (Figure 1), as will be understood by those skilled in the art.

The nature of the physical reception ports 202, 212 and physical transmission ports 206, 210 will depend upon the characteristics of the high-speed links 204, 208. For example, in the event the high-speed links 204, 208 are implemented using optical communications paths, the reception ports 202, 212 will convert optical signals received through the optical communications path into electrical signals and the transmission ports will convert electrical signals into optical signals that are then transmitted over the corresponding optical communications path.

In operation, the physical reception port 202 captures the downstream memory requests and provides the captured memory request to local hub circuitry 214, which includes control logic for processing the request and accessing the memory devices 148 over the bus system 150 to provide the corresponding data when the request packet is directed to the memory module 130a. The local hub circuitry 214 also provides the captured downstream memory request to a downstream physical transmission port 216. The downstream physical transmission port 216, in turn, provides the memory request over the corresponding downstream high-speed link 204 to a downstream physical reception port 202 in the adjacent downstream memory module 130b. The port 202 in module 130b operates in the same way as the

corresponding port in the module 130a, namely to capture the memory request and provide the request to local hub circuitry 214, which processes the request and provides the request to a downstream physical transmission port 216. The port 216 in the module 130b then operates in the same way as the corresponding port in module 130a to provide the memory request over the corresponding downstream high-speed link 204 to the next downstream memory module 130c (not shown in Figure 2).

The memory hub 140 in the module 130a further includes an upstream physical reception port 218 that receives memory responses over the corresponding upstream high-speed link 208 from an upstream physical transmission port 206 in the module 130b. The reception port 218 captures the received memory responses and provides them to the local hub circuitry 214, which stores the responses and provides the responses to the upstream physical transmission port 206, or the responses may be directed to the upstream physical transmission on the bypass bus, if the transmission port is not processing hub local responses. The upstream physical transmission port 206, in turn, provides the response over the upstream high-speed link 208 to the upstream physical reception port 212 in the system controller 212. Each of the memory modules 130 includes a corresponding downstream physical reception port 202, upstream physical transmission port 206, downstream physical transmission port 216, and upstream physical reception port 218. Moreover, these ports 202, 206, 216, 218 in each module 130 operate in the same way as just described for the corresponding ports in the module 130a.

During a synchronization mode of operation, the system controller 110 and the ports 202, 206, 216, and 218 operate to synchronize each of the downstream high-speed links 204 and upstream high-speed links 208, as will now be described in more detail with reference to Figures 3 and 4. Briefly, the synchronization process includes two stages, an initialization stage and an enablement stage. In the initialization stage, the transmission-reception port pairs associated with each high-speed link 204, 208 are synchronized, and the system controller 110 is notified when all such pairs have been synchronized. When the system controller 110 is notified the initialization stage has been completed, the enablement stage commences and the reception and

transmission ports in the controller 110 and modules 130 are sequentially enabled to start receiving and transmitting functional operations. A transmission-reception port pair is the pair of ports associated with a particular high-speed link 204, 208, and thus, for example, the ports 210 and port 202 in module 130a are a transmission-reception port pair, as are the port 216 in module 130a and the port 202 in module 130b, and so on. The link 204 or 208 and the corresponding transmission-reception port pair may be collectively referred to simply as a "link" in the following description.

Figure 3 is a functional diagram illustrating the operation of the system controller 110 and the memory modules 130a and 130b during the initialization stage of operation. In the example of Figure 3, the system memory 102 is assumed to include only the two memory modules 130a and 130b for ease of explanation, and from this example description one skilled in the art will readily understand the operation in the initialization stage when the system memory 102 includes additional memory modules. In Figure 3, the ports 210 and 212 in the system controller 110 and ports 202, 206, 216, 218 in the modules 130a, 130b have been designated using new identifiers to simplify the description of the initialization stage of operation. More specifically, the transmission port 210 and reception port 212 in the system controller 110 have been designated TXP-SC and RXP-SC, respectively, where the "SC" indicates the ports are contained in the controller and "TX" indicates a transmission port and "RX" a reception port. The downstream physical reception port 202 in modules 130a and 130b have been designated DRXP-A and DRXP-B, respectively, where the "A" and "B" indicate the ports are contained in the modules 130a and 130b, respectively, and the "U" indicates an upstream port. Similarly, the upstream physical transmission ports 206 in the modules 130a and 130b are designated UTXP-A and UTXP-B, respectively. The downstream physical transmission port 216 in module 130a is designated DTXP-A where the "D" indicates a downstream port. The upstream physical reception port 218 in module 130a is designated URXP-A. Note that since the module 130b is the last module in the example of Figure 3, the ports 216, 218 in this module are not utilized and are thus not shown in Figure 3.

To start the synchronization process, the system controller 110 and memory modules 130a, 130b are placed in an initialization mode of operation. This may occur, for example, upon power up of the computer system 100 (Figure 1). In the initialization mode, each transmission port TXP-SC, DTXP-A, UTXP-B, UTXP-A and reception port DRXP-A, DRXP-B, URXP-A, RXP-SC pair execute an initialization routine to determine a desired phase shift of a generated receive clock signal relative to test data being sent to the reception port. Thus, the TXP-SC-DRXP-A ports execute an initialization routine, as do the DTXP-A-DRXP-B 202 ports, the UTXP-B-URXP-A ports, and the UTXP-A-RXP-SC ports.

The specific initialization routine executed by each pair may vary. In one embodiment, each transmission port TXP-SC, DTXP-A, UTXP-B, UTXP-A applies test data TD to the corresponding reception port DRXP-A, DRXP-B, URXP-A, RXP-SC. The test data may have a variety of different values, and could, for example, be a pseudo random bit pattern. The reception port DRXP-A, DRXP-B, URXP-A, RXP-SC captures the test data responsive to a generated receive clock, and then determines whether the test data was successfully captured. The reception port DRXP-A, DRXP-B, URXP-A, RXP-SC adjusts the phase of the generated receive clock signal relative to the test data and once again determines whether the test data was successfully captured. In this way, the reception port DRXP-A, DRXP-B, URXP-A, RXP-SC "paints" a data eye for the test data by determining limits for the phase shift of the generated receive clock signal that allow the test data to be successfully captured. Once all phase shifts for the generated receive clock signal have been used, the reception port DRXP-A, DRXP-B, URXP-A, RXP-SC selects one of the phase shifts for use during normal operation of the system memory 102.

Because multiple high-speed links 134 must be synchronized, the controller 110 must be able to determine when all links have been successfully synchronized. Accordingly, during the initialization stage of operation, once the RXP-SC port has painted the corresponding data eye and selected the phase of the generated receive clock signal to be used during normal operation, the RXP-SC port applies an invert signal INV to the TXP-SC port. In response to the INV signal, the TXP-SC port

inverts the test data being sent to the DRXP-A port, meaning that the bit-wise complement of each test data word being transmitted is now provided by the TXP-SC port. For example, if a 15-bit pseudo random sequence "111101011001000" is applied for each bit of a test data word, the TXP-SC port would provide the complement of this sequence, namely "000101001101111." This inverted test data is indicated as TD* in Figure 3.

When the DRXP-A port receives inverted test data TD*, this indicates that the RXP-SC port has been synchronized. Once the DRXP-A port has painted the corresponding data eye and selected the phase of the generated receive clock signal to be used during normal operation (*i.e.,* has been synchronized), the DRXP-A port applies an invert signal INV to the DTXP-A port. In response to the INV signal, the DTXP-A port provides inverted test data TD* to the DRXP-B port, indicating the DRXP-A port has been synchronized. Once the DRXP-B port has been synchronized, the port applies an invert signal INV to the UTXP-B port which, in turn, applies inverted test data TD* to the URXP-A port. The URXP-A port the operates in the same way, and once synchronized applies an invert signal INV to the UTXP-A port, which then applies inverted test data TD* to the RXP-SC port.

When the RXP-SC port receives the inverted test data TD*, the system controller 110 determines that all the transmission-reception port pairs have been synchronized, and the thus all the ports are ready to be placed into a normal mode of operation to allow normal operation of the computer system 100. Accordingly, at this point the controller 110 and system memory 102 enter the enablement stage of the synchronization process, as will now be described in more detail. The term "enablement" is used to indicate that the ports are placed in a normal mode of operation to transfer or receive functional commands in the system memory 102, such as read or write commands from the system controller 110.

Figure 4 is a functional diagram illustrating the operation of the system controller 110 and memory modules 130 of Figure 2 during the enablement stage of the synchronization process according to one embodiment of the present invention. As previously mentioned, once the RXP-SC port receives the inverted test data TD* the

system controller 110 determines that all the transmission-reception port pairs have been synchronized. More specifically, when the RXP-SC port receives the TD* data, the port applies an active enable signal EN to the TXP-SC port. In response to the enable signal, the TXP-SC port is enabled as indicated by the looped arrow with the EN designation in Figure 4. Once enabled, the TXP-SC port starts sending no operation or "NOP" commands to the DRXP-A port. A NOP command is a valid command used during normal operation of a memory system but which causes a receiving module to perform no action, as will be understood by those skilled in the art.

Upon receiving the NOP commands from the TXP-SC port, the DRXP-A is enabled and also applies an active enable signal EN to the DTXP-A port. In response to the enable signal, the DTXP-A port is enabled and, in turn, starts sending NOP commands to the DRXP-B port. The DRXP-B port is enabled responsive to the NOP commands, and also provides an active enable signal EN to active the UTXP-B port. Once activated, the UTXP-B port starts sending NOP commands to the URXP-A port, and this port is enabled responsive to the NOP commands. The URXP-A port thereafter applies an active enable signal EN to the UTXP-A port to active this port, which, in turn, starts sending NOP commands to the RXP-SC port in the system controller 110. Upon receiving the NOP commands, the RXP-SC port generates a ready signal RDY, indicating that the synchronization process is now complete and signaling to the controller 110 that normal functional commands such as read and write commands may now be applied to the memory modules 130.

The initialization stage of the synchronization process synchronizes each of the links 204, 208 and the associated transmission-reception port pair. The system controller 110 is notified when all the transmission-reception port pairs have been synchronized. At this point in time, all the ports in the controller 110 and modules 130 may be enabled to allow functional commands to be processed by the system memory 102. The ports may not simply be randomly enabled, however, or erroneous operation of the system memory could result. For example, if the TXP-SC port were simply enabled once the DRXP-A port was synchronized, the controller 110 could then output a functional command through the TXP-SC port. If one of the downstream ports were

not enabled, however, then this functional command may not be applied to all memory hubs 140 as desired. For example, if the DTXP-A port was not yet enabled to transmit functional commands (*i.e.*, was still synchronizing the DRXP-B port), then the functional command would not be applied to module 130b as desired. The present synchronization process eliminates this possibility by sequentially enabling the ports in a clockwise manner starting with the TXP-SC port and ending with the RXP-SC port in the controller 110. In this way, downstream links are sequentially enabled starting from the controller 110 and progressing downstream, and upstream links are sequentially enabled starting with the module 130 furthest downstream and ending with the controller.

One skilled in the art will understand suitable circuitry for forming the components of the memory hubs 140, and will understand that the components may be implemented using either digital or analog circuitry, or a combination of both, and also, where appropriate, may be realized through software executing on suitable processing circuitry. Moreover, in the above description the ports are discussed as applying the INV and EN signals to adjacent ports during the synchronization process. These signals may be applied to the adjacent ports through the local hub circuitry 214 or directly as described, as will be appreciate by those skilled in the art. Similarly, some of functionality of the ports may be performed by the local hub circuitry 214. The division of the functionality among one or more components in the hubs 140 is not important so long as the components operate in combination to perform the described functions, as will also be appreciated by those skilled in the art.

In the preceding description, certain details were set forth to provide a sufficient understanding of the present invention. One skilled in the art will appreciate, however, that the invention may be practiced without these particular details. Furthermore, one skilled in the art will appreciate that the example embodiments described above do not limit the scope of the present invention, and will also understand that various equivalent embodiments or combinations of the disclosed example embodiments are within the scope of the present invention. Illustrative examples set forth above are intended only to further illustrate certain details of the various

embodiments, and should not be interpreted as limiting the scope of the present invention. Also, in the description above the operation of well known components has not been shown or described in detail to avoid unnecessarily obscuring the present invention. Finally, the invention is to be limited only by the appended claims, and is not limited to the described examples or embodiments of the invention.